# Server Side application Theorie

MVC : Model-View-Controller

Manier om complexe dingen te maken

DATA : Model = data klasses + business logica + data acces + validatie

GUI : View = html + css + javascript

FLOW : Controller = ontvangt of weigert de request, volgt routing regels, haalt data op, stuurt view aan

DRY : Do Not Repeet yourself

Moet goed te onderhouden zijn, makkelijk te uibreiden, kwaliteit

Conventie over configuratie

Regels voor het coderen en formateren

Regels voor eenvoud en minder discussie

Regels om sneller resultaat te hebben

Controllers

Regelen de flow v de app

Ontvangen requests

Uitlezen van input

Verwerken of doorsturen van data

Resultaat opstellen en terugsturen

Hoe kunnen we data doorgeven

ViewBag

Controller => View

QueryString

Browser => Controller

Hidden Field

Controller ⬄ View

Form

Controller ⬄ View

ViewBag

Dynamic object (opbject die geen properties heeft)

Strings, Getallen, Objecten

QueryString

Zelfde concept als PHP

Strongly Typed!!!!

Nullable parameters bv “ int? “

Model Binding

Een model wordt gekoppeld aan een formulier

Proprties worden gekoppeld via de name

Presentation Model

Klasse die een formulier nodig heeft om het te ondersteunen, voor alle data die er in moet gestoken worden

* De view bevoorraden en dergelijke

## Cookies

Controleren of cookie bestaat:

Via de http context => HttpContext.Request.Cookies[“keuze”]

Cookie wegschrijven:

HttpCookie cookie = new HttpCookie(“keuze”);

…

Response.SetCookie(cookie);

SetCookie gaat de cookie overschrijven!! Bij het andere zal hij er gewoon 1 toevoegen

machineKey wordt gebruikt om data te encrypteren en decrypteren

## Session State

Zie slides.

Session.Add(“key”,value);

## Source Control

Beheren van source code

Beheren van documenten mbt.project

Samenwerken aan projecten

## Webservices

Software die op het internet draait die je kan aanspreken van op devices.

Webservices zijn aanspreekbaar vanop het internet

Je kan van hier op je pc bv een service die op een server in amerika staat gebruiken en uitvoeren.

SOAP Simple Object Acces Protocol

Uitwisselen van gegevens, data formaat is XML

In de WSDL wordt beschreven wat je bv kan aanroepen of wat beschikbaar is en dergelijke

WCF Windows Communication Foundation

Krachtige communicatie

REST

Rest services zijn stateless

Het zijn gewoon een aantal regels die beschrijven hoe je data verstuurd over http